Annals of the Academy of Romanian Scientists
Series on Science and Technology and Information
Online ISSN **2066 - 8562** Volume **18**, Number 1/**2025** 

# VISION-BASED QUALITY CONTROL OF PRODUCTS IN DIGITAL MANUFACTURING

Theodor BORANGIU[1], Silviu RĂILEANU[2], Ionuţ LENŢOIU[3]

**Rezumat.** *Instrumentele de viziune sunt o categorie puternică de operatori de detectare și măsurare a caracteristicilor pentru controlul geometriei pieselor în sistemele Industry 4.0. Lucrarea prezintă două tipuri de instrumente de viziune pentru sarcinile de control al calității asistate de calculator în producția inteligentă: rigle liniare/circulare pentru măsurători de geometrie (lungimi și unghiuri de muchie) și detectoare de caracteristici de piesă (linii, arcuri și puncte) care găsesc segmente de contur, colţuri etc.; aceste instrumente de viziune sunt integrate în mediul de programare V+. Instrumentele de viziune proiectate funcționează atât pentru imagini în tonuri de gri, cât și pentru obiecte binare, sunt robuste pentru translația și rotirea obiectelor, deoarece pot fi legate de centrul de masă al obiectului și de axa minimă de inerție, sunt configurabile în raport cu AOI și pot fi imbricate.*

**Abstract.** *Vision tools represent a powerful category of feature detection and measuring operators for part geometry control in Industry 4.0 systems. The paper presents two types of vision tools for Computer Aided Quality Control tasks in intelligent manufacturing: linear/circular rulers for geometry measurements (edge lengths and angles), and detectors of part features (lines, arcs and points) that find contour segments, corners, a.o.; these vision tools are integrated in the V+ robot-vision programming environment. The designed vision tools work both for greyscale and binary object images, are robust to object translation and rotation because they can be referred to the object's centre of mass and minimum inertia axis, are configurable relative to the AOI and can be nested.*

**Keywords:** Geometry control, visual measurement, image processing tool, digital manufacturing

**DOI** 10.56082/annalsarsciinfo.2025.1.19

## 1. Introduction

The digital transformation of industrial processes and manufacturing reorients the automatic control of resources and inspection of products from the classic logic run on traditional programmable logic controllers and HMIs to AI-driven planning and process control strategies operating within smart semi-heterarchical super-vision systems: product-driven automation, intelligence embedded on products and in line quality inspection for zero-defect product making.

[1]University Professor, Ph.D., POLITEHNICA Bucharest, Academy of Romanian Scientists, theodor.borangiu@upb.ro
[2]Associate Professor, Ph.D., POLITEHNICA Bucharest, silviu.raileanu@upb.ro
[3]Ph.D. student, POLITEHNICA Bucharest, ionut.lentoiu@stud.acs.upb.ro

Event-driven or continuous data measurements with IoT sensors from process parameters and operations performed by resources on products, as well as in-line quality control of products assessed with computer vision serve production tasks in the cloud to extract knowledge on how shop floor processes execute, how quality is certified and defects are reduced, and how efficiency can be improved in a safe operating context. Using distributed, product-centred control methodologies these new approaches influence activities and operations in real-time. Process control is increasingly data-driven in digital manufacturing, and the capacity to exploit (big) data collected in the shop floor in real time adds value: efficiency, reality awareness and agility. New hardware and software technologies and tools like extended digital modelling, digital twins, HMI and visualisation, resource and product virtualization and predictive IA use historical data, contextualize it in the current production environment, thus ensuring global efficiency at batch level and improved situational awareness [1].

In manufacturing and logistics, different types of quality control ensure products meet required standards. These types are:

1. Pre-production inspection: Checks raw materials and initial setups before manufacturing begins.
2. In-process (in line) inspection: Monitors production to catch defects early.
3. First Article Inspection (FAI): Examines the first product to verify that production processes are correct.
4. Final inspection: Conducted on finished products before delivery.
5. Pre-shipment inspection: Ensures products meet specifications before shipping.

Each type plays a crucial role in maintaining product quality throughout the production process.

Computer vision reduces inspection time and boosts accuracy compared to manual and less advanced systems, ensuring manufacturers meet high-quality standards; existing vision-based product inspection tools seamlessly integrate with several industrial cameras and include built-in PLC communications via TCP/IP and Modbus protocols [2].

Measurements represent user-defined descriptors which either extend the set of scalar/space domain features for object recognition or operate locally, within specified areas of interest of objects, to evaluate particular geometric features of their boundary, body, or holes [3, 4].

Automated Visual Inspection (AVI) has concentrated in the first period at great extent on mass production items, as the expensive design and implementing of quality control made it unprofitable to use artificial vision techniques to other

categories of product batch sizes; however, in the last ten years, the cost of image processing hardware and software decreased continuously, which made it applicable to small batches and even one-of-a-kind, customized products [5].

There are hence new application areas for industrial vision inspection systems that have been not considered at the beginning of the last decade and which are now progressively implementing for:

- Objects made in very small quantities.
- Objects with highly complex 3-dimensional shapes: car engine blocks, castings, mouldings.
- Assemblies of components: electric motors, printed circuit boards (PCB), computer keyboards, a.o.
- Goods produced in series which are intentionally made with high individual variation, to look like being hand made.
- Objects which have a high degree of variability, such as products in food logistics (quality control, sorting, packing, storing).
- Articulated objects that are flexible or are assembled from jointed parts.
- Natural products (vegetables, fruits, meat, fishes) which that vary in size and quality and must be classified according to some criteria [6].

Automated Visual Inspection will have an increased impact on manufacturing and, processing the images of products using AI techniques will determine cost-effectiveness. AI techniques are able to detect objects perform image analysis, identify features and effect geometry measurements [7].

Computer vision held in 2022 a 3% market share in value of 0.12 billion $US in the global market value of AI in robotics, with a forecasted increase to 2 billion $US in 2032. For industrial robots with AI, machine vision includes digital image acquisition, processing, real time analysis and feature extraction to produce numeric or symbolic information such as decisions or adequate actions [8].

## 2. Geometry measurements with vision tools

There can be defined three major, closely inter-linked AVI topics under the general heading of *AI-driven measurements*:

a) *Pattern-based inspection*, to be used in case a model, graphic representation, CAD database or other document describing an item exist.

b) *Rule-based check*, that applies the expertise of a human specialist in a set of detailed rules and procedures to design and interpret the results of executing visual measurements.

c) *Nonimperative programming*, for which the operator defines the foreground elements in a part image to be detected by the vision system.

Implementing the AI techniques for measurements is based on *vision tools*, applied in specified areas- (or window regions-) of-interest – AOI (WROI). Since several different vision tools may be placed in the same area, it must be possible to predefine areas-of-interest, so that they can then be used by multiple tools.

Inspection sequences account for the same location of AOIs and for invariant sequences of measurements; this is the case when the type of objects to be inspected and their location in the image plane are a priori known.

However, the more general case is that of inspection sequences that cannot be established beforehand because either the scene's content is random or there is not a priori complete knowledge about where and how features should be detected and evaluated by objects' measurements. Obvious requirements are then for:

- mechanisms for *searching named features* (e.g., searching a frame, a corner, a crack or any other feature of interest in the object's image);
- techniques for *placing vision tools relative to* features detected at run time and/or reference frames generated dynamically by other vision tools,

which allows to easily reposition groups of tools based on new image data [9].

**Scan and search patterns**. Several *feature-search* methods are used to search features in the foreground of an image that visualized objects:

- *Farthest-feature location*: a type of search used in pattern-based inspection. Tasks of this type aim at finding the leftmost, rightmost, upmost, downmost edge point or line of an object silhouette, the location of the brightest/darkest area in an image or the position of the largest/smallest hole in a blob, the closest or farthest blob to a specified object edge.
- *Raster scan*: provides the basis of systematic searching when the position and orientation of an object's image feature is only vaguely The most used raster scan models are shown in Fig. 1.

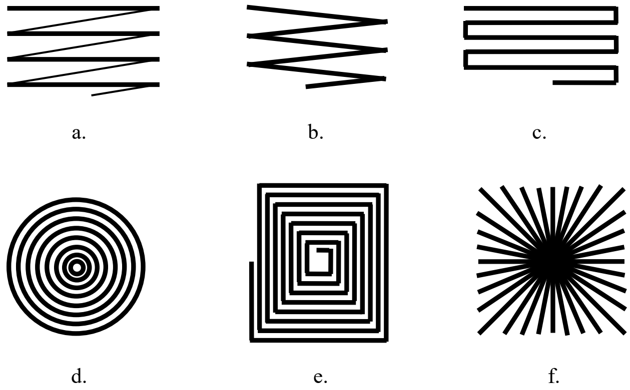


**Figure 1.** Raster scan models. a) Left-right, up-down with quick "return". b) backwards and forwards, with constant advance in the normal direction. (c) similar to (b), realizing incremental displacement along the "slow" axis. (d) concentric circular motions. (e) spiral. (f) wheel-spoke

- *Random scan*: good results at localizing big objects.
- *Space-filling scan*: this Peano scan is a pace-filling curve that preserves well clustering.
- *Coarse/fine scan search*: used when small blobs are possibly clustered in the image (e.g., defects in materials), examining objects in varying detail is the reason to adopt the coarse/fine scan search.
- *Edge/Contour following*: detects "scratch"-type features in a two-stage process of following edges, contours or streaks: i) following segments of arcs that are not interrupted (have not breaks); ii) skipping breaks, between the existing portions of an interrupted arc.
- *Navigating by landmarks*: the most common type of search pattern used in model-based inspection.

**Line and gauge predicates**. Measurements are based on the definition and selection of points, lines and arcs, to which functions are associated. Points may be selected from two categories: *predefined points* and *user points*. Predefined points are the centres of weight of objects and of holes, if any. User points can be global or local: *global points*, once defined, apply for every object whereas *local points* are defined for a particular class of objects and depend on a valid pattern database and identification process. Points can lay on the object's contour in which case they might be referred relative to the centroid of the object by polar coordinates (e.g., the nearest/farthest point on the contour relative to the centroid), or they can be related to any internal feature of the object's body in which case their offset coordinates are user-defined.

Lines and circular arcs are either primitive contour segments of the object and its holes - detected by *line* and *arc finder* tools, or specified by the user relative to important features identified at run time during object search (e.g., the major axis of the object, the edges of the minimum rectangle box, etc).

Once points, lines or arcs selected, simple patterns such as triangle, rectangle, parallelogram, and circle can be related to them to recognize additional features.

The *functions* defining measurements are based on library predicates which fall in-to two basic categories:

a) *Low-level feature recognition predicates*: they exploit the data base of points, lines and arcs that are implicitly provided by the vision system after segmenting the image and detecting objects, or explicitly defined by the user relative to important features identified at run time during object search. These predicates perform special measurements which may be added to the standard intrinsic ones (area, perimeter, holes number, eccentricity, compactness) to verify geometry details. Below are examples of such low-level predicates:
   - `angle(L1,L2)`: Compute the angle formed by lines $L_1$ and $L_2$;
   - `collinear(P,Q,R)`: Do the points P, Q, R lay on the same line?
   - `parallel([P,Q],[R,S])`: Are the lines PQ and RS parallel?
   - `perpendicular([P,Q],[R,S])`: Are PQ and RS perpendicular?
   - `in_circle(A,[C(x,y),r])`: Is A inside the circle C, r?

b) *Gauge predicates*: they are used in model-based inspection as feature-search patterns. Gauge predicates are designed specifically to perform the same kind of operations as *mechanical gauges*: caliper, ruler, compass, etc. For the examples below it is assumed that, after image binarization and segmentation, the background is white and the foreground is black.
   - `compass([X1,Y1],R,α,[X,Y])`: the predicate uses a circle of radius R, centred at $[X_1, Y_1]$. The search starting point is on the circle, at angle $\alpha$ from the horizontal (parallel to the abscissa of the vision frame). The input data when calling the predicate is $[X_1,Y_1]$, R, $\alpha$. The output data [X, Y] is the address of the first black pixel encountered as the compass moves anti-clockwise from the point $[X_1 + R\cos\alpha, Y_1 + R\sin\alpha]$, Fig. 2 left.
   - `caliper ([X1,Y1],L,α,o)`: the predicate operates similarly to a robot gripper picking an isolated object. The centre of the gripper is $[X_1,Y_1]$ and has two parallel fingers of length L. The gripper opens moving the jaws along a line inclined at angle $\alpha$ to the abscissa of the image plane. The opening of the jaws when grasping the object is o (the output data), Fig. 2 right.

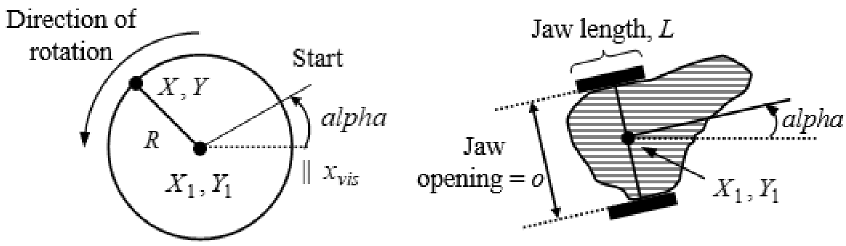


**Figure 2.** Gauge predicates: left - compass; right - caliper

**Vision Tools**. Vision tools are powerful operators which are used to detect features and measure distances along lines and arcs. They can be defined to process grayscale or binary images, after segmentation. Vision tools are employed to detect important features in an object's image: *points* (on inner (holes) and outer (contours) part edges) and *edges* (segments of lines and arcs), or to measure and locate them (line ends, point location, line inclination and arc extremities) with respect to other characteristics that have been found at run time in the image of the located object or from other previously applied vision tools.

## 3. Measuring distances, detecting points and edges on manufactured parts

Two types of part quality control (QC) using vision tools are analysed: *geometry measurements*: edge lengths and angles, hole radii relative to a) object-locating features: centre of mass (**C**), minimum inertia axis (**MIA**) or minimum rectangle box (**MRB**) edges or b) other part features (hole, corner, contour segment) already detected with some vision tools, and *detection of part features* (linear and circular contour segments, hole patterns): presence and/or correct location with respect to **C**, **MIA**, **MRB** or to other predefined and visually found anchor points [10].

These two types of part QCs are denoted respectively as **gm** and **fd** in Fig. 3, in which the methodology of selecting, configuring and using vision tools (VTs) for part control is presented. In the offline stage an image processing environment or *virtual camera* for QC is created, by customizing image representation, processing and interpreting: defining the image format (e.g., binary or grey scale, subpixel representation of VTs) and operating parameters (multiple binarization thresholds, grey level edge detectors), configuring the contrast levels, noise filters and object recognition parameters according to expected part presentation conditions (fixed / moving scenes, objects touching / overlapping, uniform / changing background, structured / unstructured foreground).

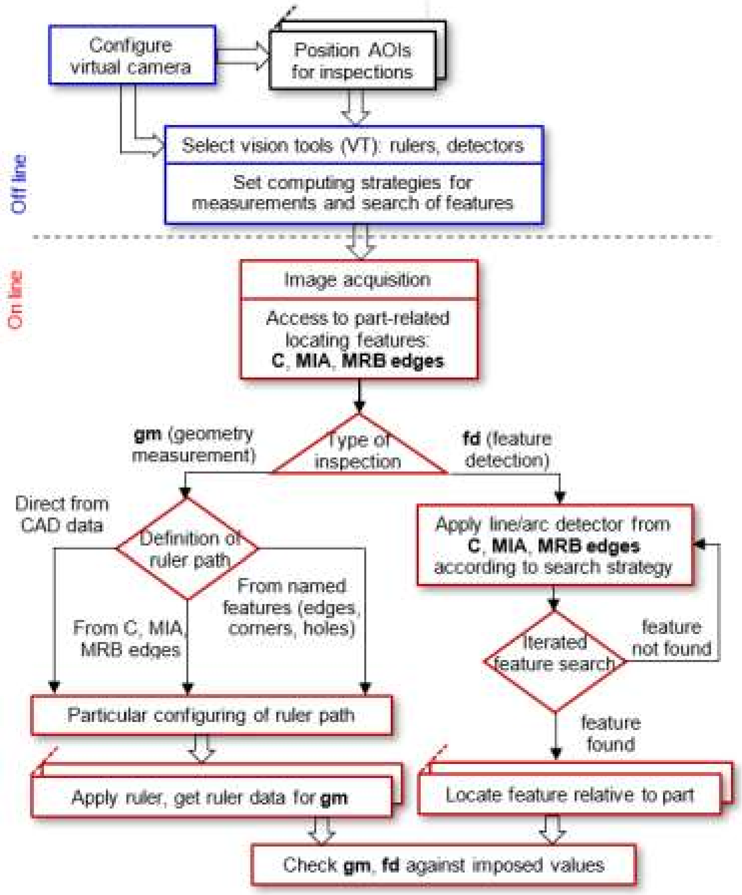


**Figure 3.** Selecting, configuring and using vision tools for part inspection: **gm** and **fd**

The spatial configuring of the image processing environment for QC consists of establishing the position (`cx`, `cy`), orientation (`ang`) and size (`wd`, `ht`) of the rectangular part inspection window or *area of interest* (AOI) for the specified virtual camera (`vc`); all measurements will be performed in the specified AOI:

```
AOI(vc) cx,cy,wd,ht,ang
```

To measure linear and circular arc distances along object and hole contours or between particular points on the object's boundary, body or holes, software rulers have been designed. Linear/circular *rulers* are vision tools that find edge (binary) information or grey levels along a line/circular arc in the current image and return such data from the start of the ruler; they can operate on binary or greyscale image data. The linear (`LRULER`) /circular (`CRULER`) ruler syntax developed for the V+ robot-vision programming environment is:

```
LRULER (vc,type,mxc,pol) data[i] = x0,y0,len,ang
CRULER (vc,type,mxc,pol) data[i] = CW,cx,cy,rad,ang0,angf
```

where: `vc` is the #ID of the virtual camera used; `type` specifies the type of ruler: (i) edge-finding rulers: dynamic binary, raw binary or run-length binary, (ii) grey level-finding rulers: grey level, fine-edge or fine-pitch; `mxc` is the max. number of values to return – the ruler stops processing upon finding this number of edges or grey levels; `pol` specifies the polarity of the edges to be considered by the ruler: only light-to-dark / all / only dark-to-light edges; `data[]`: a real array into which the edge/grey level transitions detected by the ruler are returned; the data array has the format `n,colour,value_1,…,value_n` with `n` - the number of transitions detected by the ruler, `colour` - starting colour of a binary ruler, and `value_i, i=1,…,n` are the data returned by the ruler; their interpretation depends on the ruler's type and shape:

- for a linear ruler the values are distances [mm] along the ruler from the start to each transition,
- for an arc ruler the values are angular distances [deg] from the start to each transition);

`x0,y0` indicate the coordinates [mm] of the linear ruler's starting point, `len` is the linear ruler's length [mm], `ang` is the linear ruler's angle [deg] measured counter clockwise from the vision plane abscissa *x_vis* using the start point of the ruler as origin; CW indicates how the circular ruler is moved through: clockwise / counter clockwise, `cx,cy,rad,ang0,angf` specify respectively the coordinates of the starting point [mm], the radius [mm] and the angular range [deg] of the circular ruler.

These rulers determine locations, along a directed line or circular arc, of black-to-white and/or white-to-black transitions. The starting colour returned is the colour (black or white) of the pixel nearest to the start of the ruler. Each colour change along the ruler is returned as a transition point, measured in millimetres or degrees from the ruler's start; the returned transition points are accurate within about one pixel. Rulers that operate on the greyscale object images attempt to find edges based on the setting of the edge-sensitivity threshold, previously defined in the current virtual camera, on and near the path of the ruler. These rulers look for edges based on changes in intensity rather than binary thresholded values.

The absolute speed and accuracy of rulers depend on the particular application. With the present design, ruler length and the number of transitions affect speed. Raw binary rulers are the fastest, while fine edge rulers are the most accurate. Linear rulers are faster and more accurate than arc rulers, especially when they are nearly vertical or horizontal to the vision coordinate system.

Lengths and angles measurements (**gm**) based on ruler tools need defining the ruler path, i.e., specifying the starting point, orientation, and length or angular range. The strategy for computing these data is set offline. If parts to be inspected

are in known locations of the scene, the definition of ruler parameters is direct, from CAD data. However, if the type and location of objects in the scene are not a priori known, ruler parameters can be expressed relative to a) the current position and orientation (centre of mass **C** and minimum inertia axis **MIA**) of the recognized and located object or to the edges of its minimum rectangle box **MRB** or b) to features (edges, corners, holes) visually detected in extreme positions (leftmost, upmost, …) by help of the current part locating data **C**, **MIA**, **MRB**, within the area of interest for the geometry measurement(s) of interest [11].

Part features (corners, linear and circular edges) are detected by help of *finder* tools that allow locating points, lines and arcs from **C**, **MIA** and **MRB** edge data according to the iterative search strategies and patterns (like c) in Fig. 1; they have been designed to operate on binary or raw greyscale image data. The line detector syntax developed for the V+ robot-vision programming environment is:

```
LFINDER (vc,type,pos) data[i] = xc,yc,len,wid,ang
```

where `vc` is virtual camera #ID and type selects the type of line finder: binary or grey level edge; `pos` specifies the starting point in the search window: dark side, middle guideline or light side. At the centre of the `LFINDER` search window is the guideline, i.e., the user's estimate of the edge location. The guideline is defined by a point (centre) and an angle; the search area is further specified by the guideline's length (breadth of scan), the search window's width (range about the guideline within which search will be performed), and an initial search location (see Fig. 4).

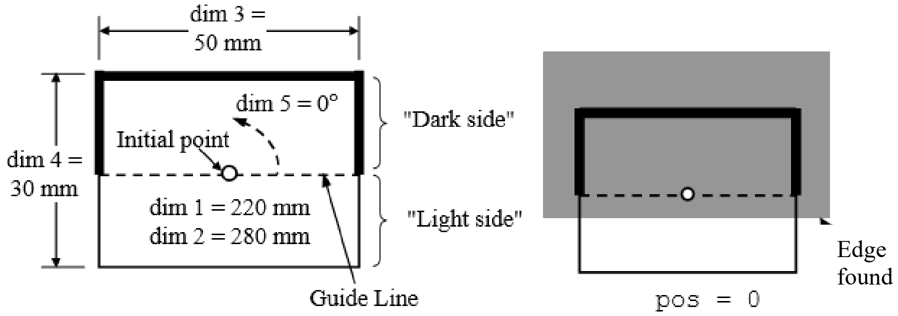


**Figure 4.** *Left*: example of line finder search area of the `LFINDER` tool. *Right*: Initial start point (start location) - middle guideline (`pos` = 0)

The result of the line fit by applying the `LFINDER` tool is put in the real array data position `data[i+0]`, while the X,Y coordinates on the fit line nearest to the initial search point [mm] and the angle of the fit line [deg] are respectively placed in `data[i+2]`, `data[i+3]`and `data[i+4]`. The line finder tool locates *dark-to-light transitions* as viewed from its dark side (see Fig. 4 right).

The point- and arc finder tools `PFINDER` and `AFINDER` are similarly defined to detect part and assembly features such as marks, corners, vertices respectively

circular holes, rings, contour segments. These types of features are related to part-attached frames and checked for correct location, size or presence (e.g., checking for missing or misaligned assembly components).

A ruler-based **gm** is further described. The dimension of a part along its minimum inertia axis – MIA) will be measured by help of a linear ruler vision tool. A window region-of-interest (`AOI`) is first placed to include the part, as indicated by the red graphics contour superposed on the object's binary image (see Fig. 5 and `V+` pseudocode below).

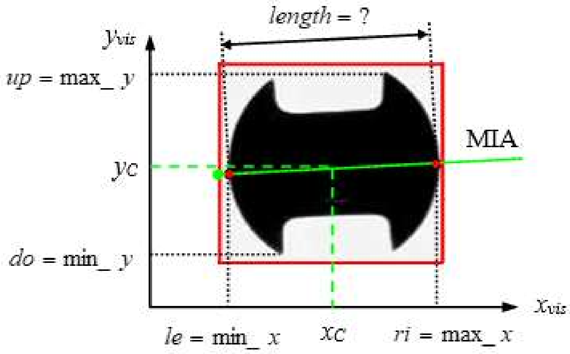


**Figure 5.** Using a linear `LRULER` to measure the part's length along its MIA

```
;Take a picture, locate the object and its minimum rectangle box (MRB).
These are implicit locating functions of any image processing library.
Access is thus granted to the part centroid coordinates C(xc,yc), angle
orient = ang(MIA,xvis), and to the MRB edge positions: le,ri,do,up.
;Compute the dimensions and centre of the minimum rectangle box:
   wid = ri – le
   ht = up – do
   cx = le + wid/2
   cy = do + ht/2
;Install the AOI processing window, with width and height 4 mm greater
;than those of the currently located object's MRB:
   AOI (vc) cx,cy,wid+4,ht+4,0
;The orientation of the MIA is computed as:
   incl = SIN(orient)/COS(orient)
;Compute the coordinates xs,ys of the linear ruler's starting point (the
;green dot in Fig. 5) by intersecting the line equations: "x = le" and
;"y – yc = incl * (x – xc)"
;The LRULER starting point coordinates result:
   xs = le
   ys = yc + incl * (le – xc)
;Launch the run-length binary linear ruler execution (type = 0) on a
;distance of 200 mm, greater than the estimated dimension to be measured
;and pol = 0 (all edges to be considered: both light-to-dark and dark-to-
;light transitions):
   LRULER (vc,0,2,2) data[] = xs,ys,200,orient
```

```
   count = data[0]
   IF  count == 2  THEN
     length = dat[3] - dat[2]
     TYPE "Part dimension = ", length
   ELSE
     TYPE "Measurement failed"
  END
;A value mxc = 2 is imposed to the binary ruler. If less than two
;transitions are found along the 200 mm MIA line and confirmed by the
system in data[0], then the measurement failed.
;"length" is the result of a successful measurement, which can be then
;checked against the imposed length.
```

The three types of vision tools (*windows* region-of interest (AOI), linear and circular *rulers*, and point-, line- and arc *finders*) can be called recursively, nested and combined in automated inspection or robot guidance vision tasks for access to products in shop floor storages or workplaces.

## 4. Experimental results

The accuracy of measurements with vision tools was analysed in experiments in most difficult conditions from the point of view of the part flow: unstructured scene foreground, parts moving on conveyor belts at different speeds, very close one to another or even touching. The tests were planned to employ all the types of ruler and finder tools designed and presented in section 3 of the paper. Tests have been performed with binary and grey level images of objects, for which accurate recognition models have been defined in preselected virtual cameras.

An example of such complex geometric measurement of a manufactured part is further presented. It describes the measurement of 10 geometric parameters of con rod-type parts. These parameters, denoted by $r, l, w_r, w_R, alpha, R, w_c, beta, h, w_e$, represent linear and angular distances (see Fig. 6).

Con rods may have any location in the inspection scene. In consequence, some important points and edges will be first searched with finder tools, and afterwards measured with ruler tools. Because the part's location is not a priori known, the first vision tools will be placed relative to part features estimated by the vision system after successfully locating the con rod. These locating features are: the coordinates of the minimum rectangle box edges, the coordinates of the part’s centroid, and the orientation of the minimum inertia axis relative to $x_{vis}$.

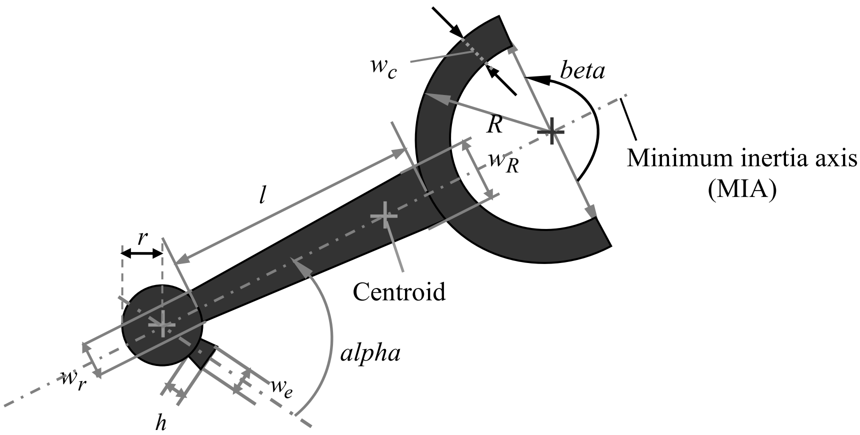


**Figure 6.** Con rod geometric parameters: $r, l, w_r, w_R, alpha, R, w_c, beta, h, w_e$.

A recognition model named "SEG" for binary images of this class of con rod parts has been previously trained. The topology of the part's contour was created by the vision system in the first stage of training (first prototype sample) with a number of 17 primitive edges (13 lines and 5 arcs), as shown in the screen capture taken during the set up of the binary recognition model (Fig. 7).

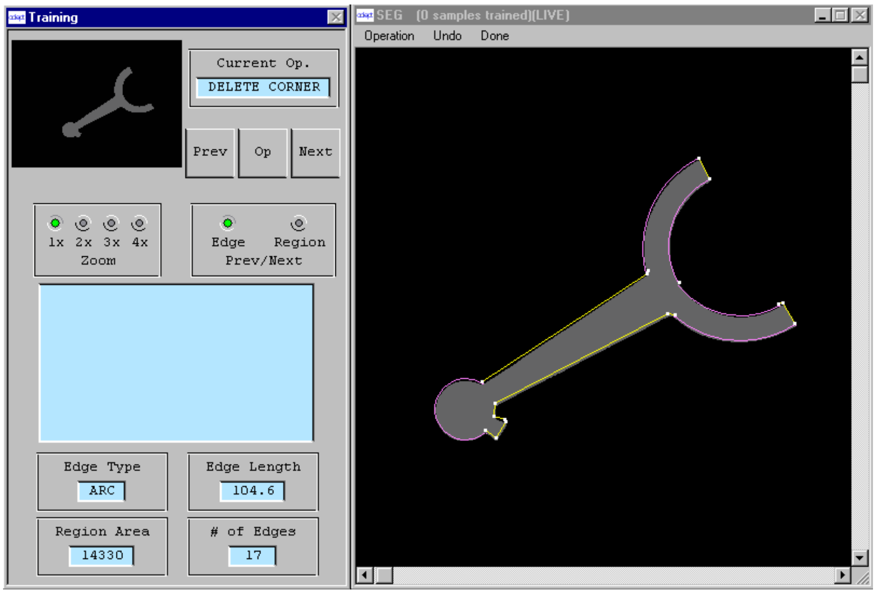


**Figure 7.** Editing the con rod recognition model (binary image prototype). Its contour is approximated with 13 lines (*yellow*) and 5 arcs (*pink*).

Eight measurements **M1**, **M2**, …, **M8** have been performed on the inspected object with different orientations. The **gm** results are given in Table 1.

**Table 1.** Results of the 8 measurement sessions with different orientations of the con rod

| **Measurement**<br>$\angle(\mathrm{MIA},x_{vis})$ [°]<br>**Parameter** [unit] | **M1**<br>3.7 | **M2**<br>36.2 | **M3**<br>91.4 | **M4**<br>161.8 | **M5**<br>181.8 | **M6**<br>222.0 | **M7**<br>270.1 | **M8**<br>328.7 |
|---|---|---|---|---|---|---|---|---|
| $r$ [mm] | 11.36 | 11.32 | 11.25 | 11.29 | 11.28 | 11.30 | 11.30 | 11.34 |
| $R$ [mm] | 35.98 | 35.74 | 35.88 | 36.07 | 35.97 | 35.90 | 35.88 | 35.95 |
| $err\_x$ [mm] | 0.10 | 0.07 | 0.01 | 0.16 | 0.27 | 0.22 | 0.07 | 0.16 |
| $err\_y$ [mm] | 0.06 | 0.12 | 0.04 | 0.10 | 0.10 | 0.03 | 0.09 | 0.08 |
| $w_c$ [mm] | 9.35 | 9.49 | 9.53 | 9.74 | 9.82 | 9.68 | 9.44 | 9.26 |
| $w_r$ [mm] | 9.55 | 9.44 | 9.02 | 9.54 | 9.46 | 9.39 | 9.12 | 9.63 |
| $l$ [mm] | 74.05 | 74.22 | 73.84 | 72.83 | 73.07 | 72.89 | 73.47 | 74.00 |
| $w_R$ [mm] | 17.73 | 18.25 | 17.70 | 17.74 | 17.65 | 17.72 | 17.65 | 17.70 |
| $beta$ [degree] | 174.6 | 176.5 | 175.6 | 174.9 | 176.0 | 174.8 | 174.2 | 174.6 |
| $alpha$ [degree] | 62.61 | 61.86 | 61.28 | 62.14 | 61.42 | 62.16 | 61.63 | 61.77 |
| $h$ [mm] | 4.13 | 4.77 | 5.15 | 4.73 | 4.59 | 4.10 | 4.06 | 4.18 |
| $w_e$ [mm] | 7.75 | 7.79 | 7.56 | 7.12 | 7.22 | 7.62 | 7.67 | 7.50 |

From the ten geometric parameters that are measured, eight are linear distances and circle radii, and two are angles.
Table 2 includes statistical data resulting from the analysis of measured values of lengths (distances and radii expressed in millimetres) and angles (in degrees).

**Table 2.** Statistic collected from the multiple con rod measurements.

| **Statistics**<br>**Parameter** [unit] | Min value<br>( min ) | Max value<br>( max ) | Mean value<br>( $avg$ ) | Dispersion<br>( $disp$ ) | Number of vision tools used | | |
|---|---|---|---|---|---|---|---|
| | | | | | RU | LF | AF |
| $r$ [mm] | 11.25 | 11.36 | 11.30 | 0.11 | 1 | 0 | 1 |
| $R$ [mm] | 35.74 | 36.07 | 35.92 | 0.33 | 3 | 0 | 3 |
| $w_c$ [mm] | 9.26 | 9.82 | 9.54 | 0.52 | 3 | 0 | 3 |
| $w_r$ [mm] | 9.02 | 9.63 | 9.39 | 0.61 | 5 | 0 | 3 |
| $l$ [mm] | 72.83 | 74.22 | 73.54 | 1.39 | 6 | 0 | 3 |
| $w_R$ [mm] | 17.65 | 18.25 | 17.75 | 0.60 | 8 | 0 | 3 |
| $beta$ [degree] | 174.2 | 176.5 | 175.1 | 2.3 | 8 | 2 | 3 |
| $alpha$ [degree] | 61.28 | 62.61 | 61.73 | 1.33 | 9 | 2 | 3 |
| $h$ [mm] | 4.06 | 5.15 | 4.46 | 1.09 | 10 | 2 | 3 |
| $w_e$ [mm] | 7.12 | 7.79 | 7.53 | 0.67 | 13 | 2 | 3 |

Table 2 shows the min., max., and mean values of the 10 measured parameters, and the dispersion of values for each parameter. The dispersion is calculated for each parameter $P_i, i = 1, \ldots, 10$ as: $disp(P_i) = max(P_i) - min(P_i)$, and is expressed in the same units as the parameter (millimetres or degrees).
Table 2 also specifies the number of vision tools (*rulers* – RU, *line finders* – LF, *and arc finders* – AF) that are used by the vision program to obtain the value of each geometric parameter.
Fig. 8 offers an interpretation of the statistical results obtained in the measuring process of the 8 con rod's linear distances. The graphic representation of feature dispersion is a function of the feature's length [millimetres] and number of vision tools (rulers, line and arc finders) that have been used.

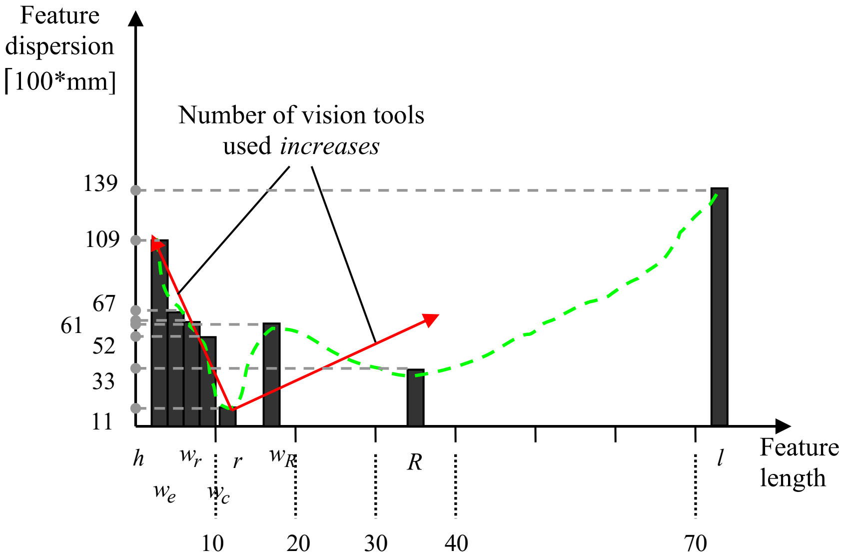


**Figure 8.** Graphic representation of feature dispersion function of the feature's length [millimetres] and number of vision tools (rulers, line and arc finders) used in the measuring program

One can observe from this representation that the feature dispersion curve has a minimum in the range of medium feature lengths ($9 \ldots 40$ millimetres, parameters $R, r, w_c$) and also for those feature parameters that require the least number of vision tools (RU, LF, and AF) to be executed. The same three con rod parameters mentioned above meet this second computational condition: $R(\text{RU} = 3, \text{LF} = 0, \text{AF} = 3)$, $r(1,0,1)$, $w_c(3,0,3)$. Looking at the values in the last three columns of Table 2, it results that the con rod parameters $R, r, w_c$ are calculated first, while for the remaining five distance-features computing errors cumulate progressively as new vision tools are executed by nested measuring programs.

A final note concerns the dependence of measuring accuracy on the feature's length. The results worsen as the length of the search window increases. To exemplify, the relative dispersion $p\_disp(P_i) = \frac{disp(P_i)}{avg(P_i)} \cdot 100$ [%], defined as a percentage of the feature's mean value, is computed for $r$ and $l$, i.e., for the parameters having respectively the smallest and biggest absolute dispersion of their measuring results: $p\_disp(r) = 0.97\ \%$, and $p\_disp(l) = 1.89\ \%$. While $l$ is 6.5 times longer than $r$, their relative dispersions are in the ratio of only 1.9.

## 5. Conclusions

Vision systems in the Industry 4.0 (I4.0) perspective automate in line geometry inspection of products within the CAQC (computer-aided quality control) stage of intelligent manufacturing. CAQC applications span a broad range of applications from simple checking of missing and alignment of assembly components to real time feature detection and high precision geometry measurements [12].

Artificial vision technologies are increasingly used in the I4.0 context for high-speed production and internal logistics processes; the ultimate goal is to get zero-defect production performance, plus relaxing the constraints of rigid structuring of part transport and storage flows [13, 14].

The paper proposes two types of vision tools for CAQC tasks: linear/circular rulers for geometry measurements (edge lengths and angles), and detectors of part features (lines, arcs and points) that find contour segments, corners, a.o.; these vision tools are integrated in the V+ robot-vision programming environment. The designed vision tools work both for greyscale and binary object images, are robust to object translation and rotation because they can be referred to the object’s centre of mass and minimum inertia axis, are configurable relative to the AOI and can be nested.

These vision tools have been tested in an industrial environment for binary- and greyscale object recognition models; they offer a subpixel accuracy that has been attested in repeated sequences of multiple part geometry measurements.

Future research will be oriented towards developing quality control algorithms for the complex geometry of domain features of particular object classes: skeleton and signatures (angular radii and linear offsets).

## R E F E R E N C E S

[1] IBM, Document Analytics. Visual Recognition Classifier actions, IBM Data Cap 9.1.7, IBM Watson, https://www.ibm.com/docs/en/datacap/9.1.7?topic=actions-visualrecogclassify, accessed 25 October 2024

[2] Autmix, Computer vision for quality control, 2024, https://autmix.com/en/blog/computer-vision-quality-control
[3] Melissa, R., Robots everywhere, 2nd edition, A Comprehensive Market Guide for Robotics Enthusiasts, 2nd Edition, 2023, Statzon
[4] SICK, PLB 5.7 Robot Guidance Systems, Operating instructions, https://www.sick.com/ag/en/system-solutions/robot-guidance-systems/plb/c/g259663 Accessed on 25 October 2024
[5] Cognex, 3D-A5000 Series Area Scan 3D Camera, https://www.cognex.com/products/machine-vision/3d-machine-vision-systems/3d-a5000-series-area-scan , Accessed on 3 November 2024
[6] Răileanu, S., Anton, F., Borangiu, T., Anton, S., Using the Digital Twin Concept to Design and Validate a Robotized Bin Picking Process. In: Zeghloul S., Laribi M.A., Sandoval J. (eds) Advances in Service and Industrial Robotics. RAAD 2021. Mechanisms and Machine Science, vol 102. pp. 266-274, 2021, ISBN 978-3-030-75258-3, https://doi.org/10.1007/978-3-030-75259-0_29, Springer, Cham
[7] Quinn, A., AI Image Recognition: Inevitable Trending of Modern Lifestyle, Top Ten AI, Internet archive, 2022, https://topten.ai/ai-image-recognition/
[8] Munde, S., Robotic Vision Market Research Report, Market Research Future, 2024, document MRFR/SEM/1317-HCR, https://www.marketresearchfuture.com/reports/robotic-vision-market-1849#
[9] Corke, P., Jachimczyk W., Pillat R., Robotics, Vision and Control: Fundamental Algorithms In MATLAB, 3e, 2023, ISBN: 9783031072611, Springer International Publishing
[10] Cognex, Vision Tools, 2025, https://www.chromos.ch/industrial/wp-content/uploads/sites/4/2022/10/Product_Guide_Vision_Tools_EN.pdf
[11] Keyence, Vision-Guided Robotics, WW-GB 2084-1 601A07, 2024, Keyence Corp.
[12] Konstantinidis, F.K., Mouroutsos, S.G., Gasteratos, A., The Role of Machine Vision in Industry 4.0: an automotive manufacturing perspective, IEEE International Conference on Imaging Systems and Techniques (IST), pp. 1–6, 2021, https://doi.org/10.1109/IST50367.2021.9651453
[13] Quinn, A., AI Image Recognition: Inevitable Trending of Modern Lifestyle, Top Ten AI, 2022, Internet archive, https://topten.ai/ai-image-recognition/
[14] Psarommatis, F., May, G., Azamfirei, V., Konstantinidis, F., Optimizing efficiency and zero-defect manufacturing with in-process inspection: challenges, benefits, and aerospace application, Procedia Computer Science, Vol. 232, pp. 2857-2866, 2024, https://doi.org/10.1016/j.procs.2024.02.102